72. Postman and httpGet, httpPost, httpPut, httpDelete Method - 19 July 2022

Today we will learn:

1] Call httpGet, httpPost, httpPut, httpDelete Method using PostMan

httpGet 🡺 To Fetch the record from Target Org

1] Fetch 10 Accounts from Target Org.



|  |  |
| --- | --- |
| SOURCE ORG (POSTMAN) | TARGET ORG |
|  |  |

2] Postman sends Account Name = “Cinemax” and fetch the related data from Target Org

@RestResource(urlMapping='/getAllAccountData')

global class AccountProvider {

@httpGet

global static Account getSelectedAccounts(){

RestRequest req = RestContext.request;

RestResponse res = Restcontext.response;

String jsonString=req.requestBody.tostring();

System.debug('Received Request Data = '+jsonString);

responseWrapper objResWrap=(responseWrapper) JSON.deserialize(jsonString,responseWrapper.class);

return ([select Id, Name, Type, Rating, CreatedDate,SLA\_\_c from Account where Name=: objResWrap.Name LIMIT 1]);

}

global class responseWrapper{

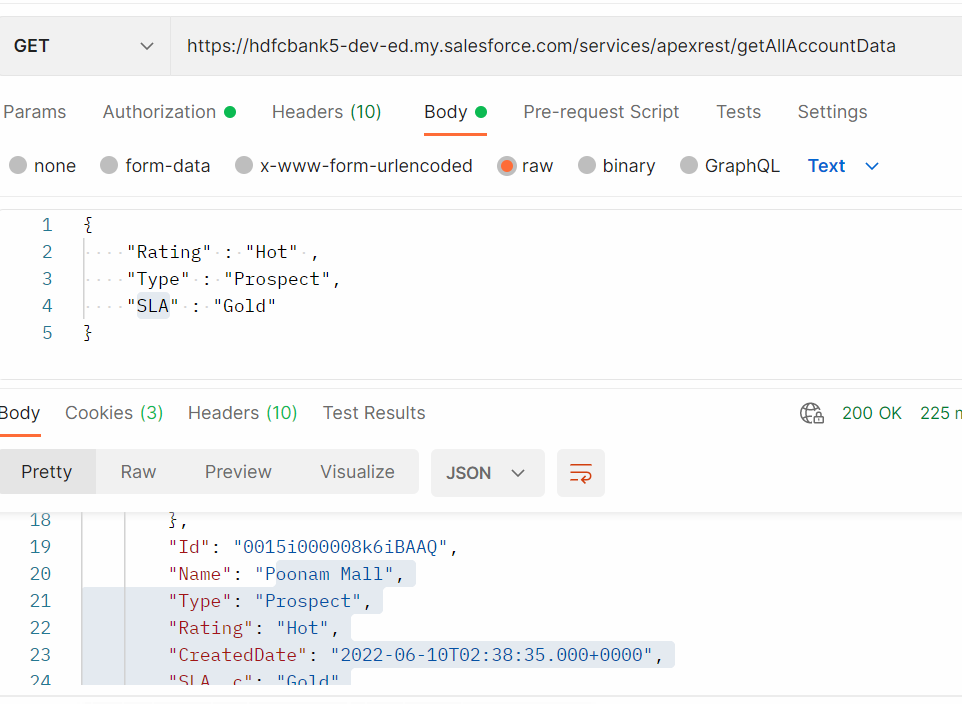
global string name;

}

}

|  |  |
| --- | --- |
| POSTMAN | TARGET ORG |
|  |  |

2] Send Rating and Type from Postman and fetch the related records from Target Org.



@RestResource(urlMapping='/getAccountDetails')

global class AccountProvider {

@httpGet

global static List<Account> getAccountDetails(){

RestRequest request = RestContext.request;

RestResponse response = RestContext.response;

String receivedData = request.requestBody.toString();//Type, Rating, SLA

AccountProviderWrapper objAccProviderWrapper = (AccountProviderWrapper) JSON.deserialize(receivedData, AccountProviderWrapper.class);

return [select Id, Name, Type, Rating, SLA\_\_c, CreatedDate from Account where Type=: objAccProviderWrapper.type and Rating=:objAccProviderWrapper.rating and SLA\_\_c=:objAccProviderWrapper.sla];

}

global class AccountProviderWrapper{

global String type;

global String rating;

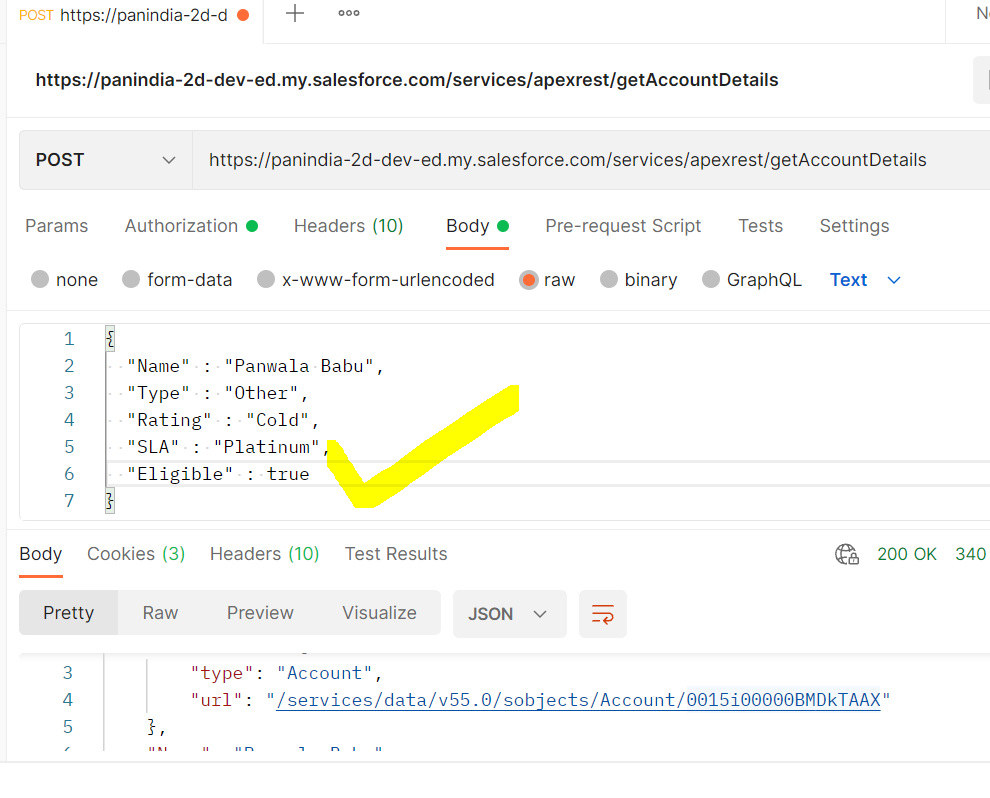
global String sla;

}

}

httpPost 🡺 To Create a new Record in the Target Org

1] Send Account Name, Rating, SLA from Postman and create this record on Target ORG.



CODE:

@RestResource(urlMapping='/getAccountDetails')

global class AccountProvider {

@httpPost

global static Account getAccountDetails(){

RestRequest request = RestContext.request;

RestResponse response = RestContext.response;

String receivedData = request.requestBody.toString();//Name, Type, Rating, SLA

AccountProviderWrapper objAccProviderWrapper = (AccountProviderWrapper) JSON.deserialize(receivedData, AccountProviderWrapper.class);

Account objAcc = new Account();

objAcc.Name = objAccProviderWrapper.name;

objAcc.Type = objAccProviderWrapper.type;

objAcc.Rating = objAccProviderWrapper.rating;

objAcc.SLA\_\_c = objAccProviderWrapper.sla;

objAcc.Eligible\_\_c = objAccProviderWrapper.eligible;

insert objAcc;

return objAcc;

}

global class AccountProviderWrapper{

global String name;

global String type;

global String rating;

global String sla;

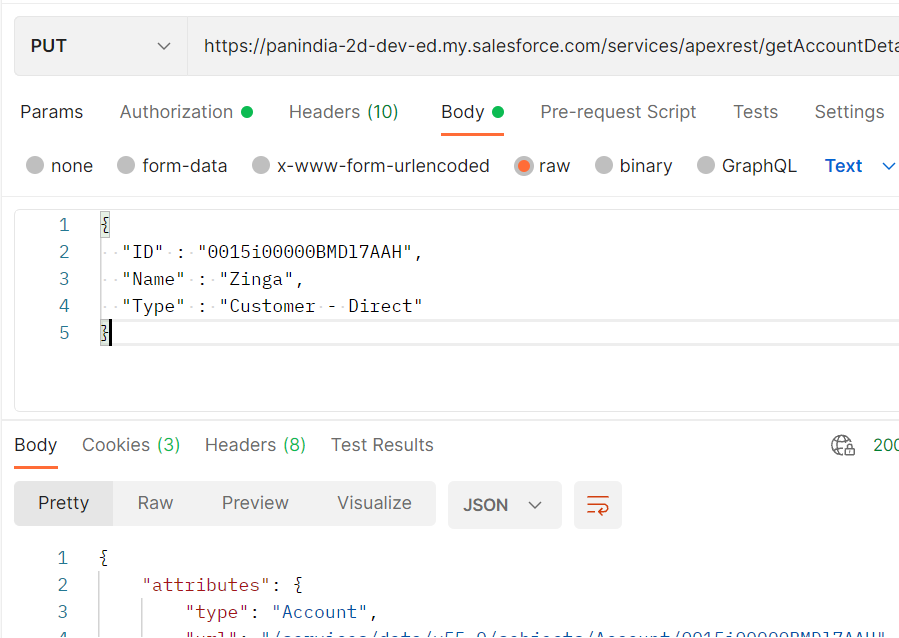
global Boolean eligible;

}

}

httpPut 🡺 To Full Update the Record on the Target

User Story : Update Account Name “Panwala Babu123” to “Zinga” and Type to “Customer - Direct”.



@RestResource(urlMapping='/getAccountDetails')

global class AccountProvider {

@httpPut

global static Account getAccountDetails(){

RestRequest request = RestContext.request;

RestResponse response = RestContext.response;

String receivedData = request.requestBody.toString();//Id, Name, Type

AccountProviderWrapper objAccProviderWrapper = (AccountProviderWrapper) JSON.deserialize(receivedData, AccountProviderWrapper.class);

Account objAcc = [select Id, Name, Type from Account where Id=: objAccProviderWrapper.id LIMIT 1];

objAcc.Name = objAccProviderWrapper.name;

objAcc.Type= objAccProviderWrapper.type;

Database.update(objAcc , false);

return objAcc;

}

global class AccountProviderWrapper{

global Id id;

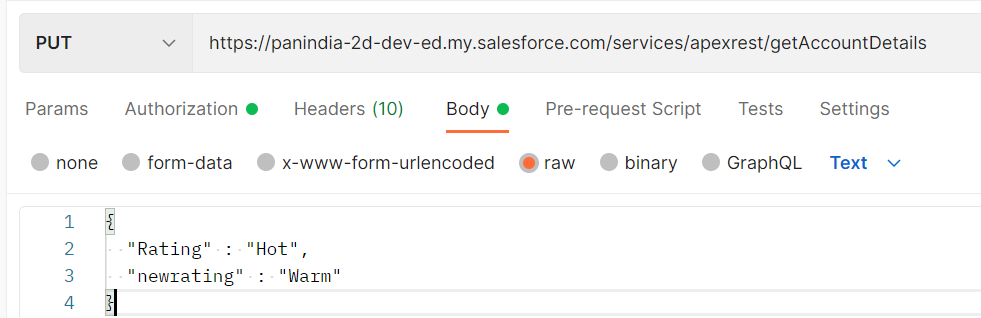
global String name;

global String type;

}

}

User Story : Update All the Account Rating to “Warm”, if Rating is “Hot”.



@RestResource(urlMapping='/getAccountDetails')

global class AccountProvider {

@httpPut

global static List<Account> getAccountDetails(){

RestRequest request = RestContext.request;

RestResponse response = RestContext.response;

String receivedData = request.requestBody.toString();//Rating, newrating

AccountProviderWrapper objAccProviderWrapper = (AccountProviderWrapper) JSON.deserialize(receivedData, AccountProviderWrapper.class);

List<Account> accList = [select Id, Name, Rating from Account where Rating=:objAccProviderWrapper.rating];//Hot

for(Account objAcc : accList){

objAcc.Rating = objAccProviderWrapper.newRating;

}

Database.update(accList, false);

return accList;

}

global class AccountProviderWrapper{

global Id id;

global String name;

global String type;

global String rating;

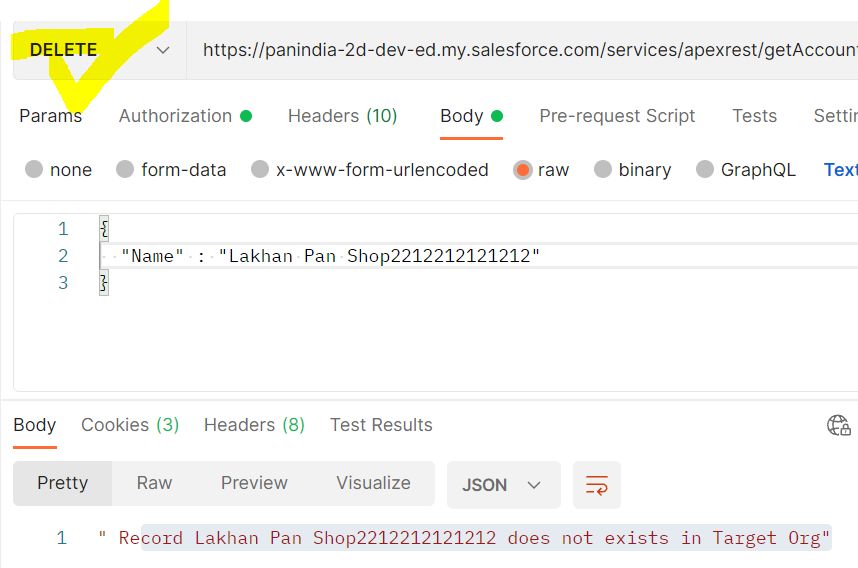
global String newRating;

}

}

httpDelete 🡺 To delete a specific record from the Target Org

User Story 1] Send Account Name from Postman, and delete this record from Target Org, if exists. Else give message, that this record does not exists.



@RestResource(urlMapping='/getAccountDetails')

global class AccountProvider {

@httpDelete

global static String getAccountDetails(){

RestRequest request = RestContext.request;

RestResponse response = RestContext.response;

String receivedData = request.requestBody.toString();//Name

AccountProviderWrapper objAccProviderWrapper = (AccountProviderWrapper) JSON.deserialize(receivedData, AccountProviderWrapper.class);

try{

if(String.isNotBlank(objAccProviderWrapper.name)){

Account objAcc = [select Id from Account where Name=: objAccProviderWrapper.name];

if(objAcc !=null){

Database.delete (objAcc , false);

return 'Record '+objAccProviderWrapper.name+' has been deleted successfully...!!!';

}

else{

return ' Record '+objAccProviderWrapper.name+' does not exists in Target Org';

}

}

else{

return ' Record '+objAccProviderWrapper.name+' does not exists in Target Org';

}

}

catch(Exception ex){

return ' Record '+objAccProviderWrapper.name+' does not exists in Target Org';

}

}

global class AccountProviderWrapper{

global Id id;

global String name;

global String type;

global String rating;

global String newRating;

}

}